SIGNAL-02

Disable signals before executing setuid(root).

Sean Barnum, Cigital, Inc. [vita[1]]

2007-04-16

Part "Original Cigital Coding Rule in XML"

Mime-type: text/xml, size: 4314 bytes

Attack Category	- Identity Spoofing - Privilege Exploitation
Vulnerability Category	- Privilege escalation problem - Process management
Software Context	- Process Management - Debug API
Location	- signal.h
Description	Signal handlers run at the privilege of the owning process. Therefore, if a process is currently running in setuid(root) when a signal fires, the signal will be operating as root privilege. "This signal() facility is a simplified interface to the more general sigaction(2) facility. Signals allow the manipulation of a process from outside its domain as well as allowing the process to manipulate itself or copies of itself (children). There are two general types of signals: those that cause termination of a process and those that do not. Signals which cause termination of a program might result from an irrecoverable error or might be the result of a user at a terminal typing the `interrupt ' character. Signals are used when a process is stopped because it wishes to access its control terminal while in the background (see tty(4)). Signals are optionally generated when a process resumes after being stopped, when the status of child processes changes, or when input is ready at the control terminal. Most signals result in the termination of the process receiving them if no action is taken; some signals instead cause the process receiving them to be stopped, or are simply discarded if the process has not requested otherwise. Except for the SIGKILL and SIGSTOP signals, the signal() function allows for a signal to be caught, to be ignored, or to generate an interrupt."

[1] http://buildsecurityin.us-cert.gov/bsi/about_us/authors/35-BSI.html (Barnum, Sean)

<table>
<tr><td></td><td>- man page for signal(3), BSD Library Functions Manual
See signal(7) for comprehensive list of supported signals.
Disable signals before executing setuid(root).
Tag any instances of setuid(root).
Warn the user to disable signals prior to executing setuid(root).</td></tr>
<tr><td><b>APIs</b></td><td><table><tr><th>Function Name</th><th>Comments</th></tr><tr><td>setuid</td><td>look for nearby instances of signals</td></tr></table></td></tr>
<tr><td><b>Method of Attack</b></td><td>An attacker may be able to hijack a low-privileges signal handler and then force execution of a that signal from a process running as root and thereby gain inappropriate privileges on the system.</td></tr>
<tr><td><b>Exception Criteria</b></td><td></td></tr>
<tr><td><b>Solutions</b></td><td><table><tr><th>Solution Applicability</th><th>Solution Description</th><th>Solution Efficacy</th></tr><tr><td>Always.</td><td>General guidance is to disable signals before executing setuid(root). Re-enable them when finished.</td><td>Effective.</td></tr></table></td></tr>
<tr><td><b>Signature Details</b></td><td></td></tr>
<tr><td><b>Examples of Incorrect Code</b></td><td></td></tr>
<tr><td><b>Examples of Corrected Code</b></td><td></td></tr>
<tr><td><b>Source Reference</b></td><td>• NetBSD Library Functions Manual. SIGNAL(3)[2] (2004).</td></tr>
<tr><td><b>Recommended Resource</b></td><td></td></tr>
<tr><td><b>Discriminant Set</b></td><td><table><tr><td><b>Operating System</b></td><td>• Windows</td></tr><tr><td><b>Languages</b></td><td>• C
• C++</td></tr></table></td></tr>
</table>

Cigital, Inc. Copyright

The Build Security In (BSI) portal is sponsored by the U.S. Department of Homeland Security (DHS), National Cyber Security Division. The Software Engineering Institute (SEI) develops and operates BSI. DHS funding supports the publishing of all site content.

[1] mailto:copyright@cigital.com